1. A method of operating a re-programmable non-volatile memory system having its memory cells organized into distinct blocks of simultaneously erasable cells, comprising: designating a first group of said blocks for storing user data and a second group of said blocks for storing information of the characteristics of said first group of blocks, storing, in individual ones of the first group of said blocks, user data plus characteristics of the user data being written therein but not including characteristics of said first group of blocks, and storing, in individual ones of the second group of said blocks, a plurality of records of characteristics of individual ones of the first group of blocks but without storing either user data or characteristics of the user data into the second group of blocks.

2. The method of claim 1, wherein storing the user data characteristics in individual ones of the first group of blocks includes storing redundancy codes generated from the user data stored therein.

3. The method of claim 1, wherein storing the plurality of records in individual ones of the second group of said blocks includes storing a redundancy code generated from records written therein.

4. The method of claim 1, wherein storing, in the second group of blocks, the plurality of records of characteristics of the first group of blocks individually includes storing programming and reading characteristics of a corresponding one of the first group of blocks.

5. The method of claim 4, additionally comprising: reading the records from a plurality of said second blocks and storing the read records in a controller memory, and when accessing one or more of the first group of blocks to program user data therein or to read user data therefrom, reading from the controller memory those of the records stored therein which contain the characteristics of said one or more of the first group of blocks being accessed.

6. The method of claim 5, wherein records of at least one of the plurality of second blocks stored in the controller memory which have the longest time since being read are removed therefrom when a limited capacity of the controller memory requires space to be made for records from another of the plurality of said second blocks to be stored therein in order to be read when one or more of corresponding ones of the first group of blocks is being accessed.

7. The method of claim 5, wherein, when a plurality of the first group of blocks with successive addresses are being accessed, an address of a record stored in the controller memory that corresponds to a first of the addressed block within the first group of blocks is calculated and remaining records within the controller memory that correspond to others of the plurality of the first group of blocks being accessed are addressed by incrementing from one record address to another.

8. The method of claim 1, wherein storing, in the second group of blocks, the plurality of records of characteristics of the first group of blocks individually includes storing an indication of whether a corresponding block within

said first group is defective or not, and, if so, storing an address of a substitute block, and, if not, storing operating characteristics of the corresponding block within said first group.

9. The method of claim 8, wherein storing operating characteristics includes storing any of programming, reading, erase or wear characteristics of the corresponding block within the first group.

10. The method of claim 1, wherein storing, in the second group of blocks, the plurality of records of characteristics of the first group of blocks individually includes storing indications of locations of any bad columns that extend through corresponding blocks within said first group.

11. The method of claim 10, wherein storing user data into individual ones of the first group of blocks and storing block characteristic records into individual ones of the second group of blocks includes skipping any bad column locations in the respective blocks.

12. The method of claim 1, wherein storing user data in individual ones of the first group of said blocks includes simultaneously writing user data into a plurality of the first group of blocks until at least one user data sector is written into each of the plurality of blocks in the first group, the individual data sectors including at least one characteristic of its user data.

13. The method of claim 12, wherein said at least one characteristic of the user data that is included as part of sectors of data includes redundancy codes that have been generated from user data while the user data is being transferred in a stream to said individual blocks within said first group, individual ones of the redundancy codes being appended to ends of the user data from which they are generated to form sectors of data.

14. The method of claim 13, wherein the redundancy codes are generated in a single circuit used for all of the plurality of the first group of blocks to which user data are simultaneously being transferred.

15. The method of claim 12, wherein said at least one characteristic includes one or more bits by which the user data of within a data sector is transformed before being stored.

16. The method of claim 12, wherein said at least one characteristic includes a plurality of bits of varying values independent of the user data and which are used to determine whether a sector of data including said plurality of bits is stored in respective blocks within said first group.

17. The method of any one of claim 1, 4, 8 or 12, wherein the memory cells within at least a plurality of said blocks are individually operated with more than two storage states in order to store more than one bit of data per memory cell.

18. The method of any one of claim 1, 4, 8 or 12, wherein the memory cells within at least a plurality of said blocks are individually operated with exactly two storage states in order to store exactly one bit of data per memory cell.

19. The method of any one of claim 1, 4, 8 or 12, wherein the memory cells within a first group of a plurality of said blocks are individually operated with more than two storage states in order to store more than one bit of data per memory cell, and wherein the memory cells within a second group of a plurality of said blocks different from said first group are individually operated with exactly two storage states in order to store exactly one bit of data per memory cell.

20. A method of managing a non-volatile flash memory system having its memory cells organized into distinct blocks of simultaneously erasable cells, comprising: storing within individual ones of said blocks at least one sector of data including user data, a redundancy code generated from the user data and a plurality of bits at a beginning of the sector that define a function by which the user data and redundancy code are transformed prior to being stored, and storing at least one characteristic of said individual ones of said blocks in at least one block other than said individual ones of said blocks, said at least one other block storing said at least one characteristic of a plurality of said blocks storing user data.

21. The method of claim 20, wherein said at least one characteristic includes an indication of whether a corresponding one of said individual ones of said

blocks is defective or not, and, if so, an address of a substitute block, and, if not, programming characteristics of the corresponding block.

22. A memory system, comprising: an array of non-volatile memory cells formed in blocks of cells that are simultaneously erasable together, a first plurality of the blocks being designated to store user data and a second plurality of the blocks being designated to individually store records of characteristics of first plurality of blocks, a controller memory separate from the array of non-volatile memory cells in which at least some of the records from the second plurality of the blocks are temporarily stored, said controller memory being characterized by having a faster access time than that of the non-volatile memory cell array, and a controller adapted to communicate sectors of user data between a host and the first plurality of the memory cell blocks while utilizing records in the controller memory from the second plurality of blocks that correspond to those of the first plurality of blocks with which user data are communicated.

23. The memory system of claim 22, wherein the memory cell array and the controller are enclosed in a card having electrical contacts thereon that match electrical contacts of a socket of a host system, the card thereby being removably connectable with the host.

24. The memory system of claim 22, wherein the memory cell array and the controller are embedded within a package containing the host system.

25. A memory system, comprising:

at least two non-volatile memory cell arrays formed on at least two respective integrated circuit chips, wherein the memory cells of each of the memory cell arrays are grouped into a number of blocks designated to individually store a given quantity of user data, and further wherein the number of such available blocks is different in individual ones of said at least two memory cell arrays, a memory controller, and a record stored in the memory system which contains non-overlapping logical address assignments of the blocks of each of the memory cell arrays, thereby to allow the controller to determine from a logical block address which of the memory arrays a corresponding physical block lies.

26. The memory system of claim 25, wherein the logical address assignment record is stored within one of said at least two integrated circuit chips.

27. The memory system of claim 25, wherein said at least two integrated circuit chips and the controller are positioned within an enclosed memory card having electrical contacts thereon that match electrical contacts of a socket of a host system, the card thereby being removably connectable with the host.

28. The memory system of claim 25, wherein said at least two integrated circuit chips and the controller are embedded within a host system.

29. The memory system of claim 25, wherein the controller is formed on one of said at least two integrated circuit chips.

30. The memory system of claim 25, wherein the controller is formed on an integrated circuit chip without a non-volatile memory cell array and that is in addition to said at least two integrated circuit chips.

31. A method of manufacturing a memory system, comprising: installing and interconnecting at least first and second integrated circuit chips that individually include an array of non-volatile memory cells, wherein said at least first and second integrated circuit chips individually contains stored therein a record of at least a number of blocks of capacity of its memory cell array for storing user data, and merging the memory array capacity records of each of said at least first and second integrated circuit chips to form a merged record on said first integrated circuit chip of contiguous ranges of logical memory block addresses assigned to the memory cell arrays of each of the at least first and second integrated circuit chips.

32. The method of claim 31, wherein the number of blocks of memory capacity for storing user data is different among said at least first and second circuit chips.

33. The method of claim 31, wherein the number of blocks of memory capacity for storing user data is the same among said at least first and second circuit chips.

34. The method of claim 31, additionally comprising installing said at least first and second circuit chips within an enclosed memory card having electrical contacts thereon for engaging contacts of a host connector.

35. The method of claim 31, additionally comprising embedding said at least first and second circuit chips within a package containing the host system.

36. A method of operating a flash EEPROM system having its memory cells organized into distinct blocks of a number of simultaneously erasable cells capable of storing a given quantity of data, comprising: providing the memory system with a controller and a plurality of physically distinct arrays of said memory cells that are individually organized into said memory blocks, storing in one of the memory blocks a record of a number of blocks available in each of said plurality of memory cell arrays for storing user data and non-overlapping ranges of contiguous logical addresses assigned to said number of user data blocks of the individual memory cell arrays, and locating a physical address of a memory cell block at least in part by accessing the record with a logical memory cell block address in order to determine one of the plurality of memory cell arrays in which the addressed memory cell block resides.

37. The method of claim 36, wherein the number of blocks of memory capacity for storing user data is different among at least two of the plurality of memory cell arrays.

38. The method of claim 36, wherein the number of blocks of memory capacity for storing user data in each of the plurality of memory cell arrays is the same.

39. The method of claim 36, wherein the plurality of memory arrays are enclosed in a memory card having electrical contacts thereon for engaging contacts of a host connector.

40. The method of claim 36, wherein the plurality of memory arrays are embedded within a package containing a host system.

41. The method of claim 36, wherein the memory cells within at least some of said memory cell blocks are individually operated with more than two storage states in order to store more than one bit of data per memory cell.

42. The method of claim 36, wherein the memory cells within at least some of said memory cell blocks are individually operated with exactly two storage states in order to store exactly one bit of data per memory cell.

43. A method of operating a re-programmable memory system having non- volatile memory cells organized into distinct blocks of a number of simultaneously erasable cells capable of storing a given quantity of data, the blocks of cells being further organized into a plurality of units, comprising: receiving and temporarily storing at least a given number of sectors of user data to be programmed

into the memory system, simultaneously programming a chunk of user data from each of the given number of temporarily stored sectors of user data into different blocks of memory cells within a number of different memory cell units equal to said given number, each chunk being a fraction of a sector of user data equal to one-half or less, and repeating the simultaneous programming of chunks of user data until all the data of each of the given number of temporarily stored sectors has been programmed into the different blocks within the given number of memory cell units.

44. The method of claim 43, additionally comprising, prior to programming chunks of data into blocks of memory cells, alternately transferring one chunk at a time in sequence from the sectors of temporarily stored user data into a plurality of storage registers equal to said given number, and thereafter the programming includes transferring the chunks of data stored in the storage registers in parallel into the blocks of memory cells within the given number of units.

45. The method of claim 44, wherein each of the given number of sectors of user data is received and temporarily stored before commencing to transfer chunks thereof into the storage registers.

46. The method of claim 44, wherein only a portion of each of the given number of sectors of user data is received and temporarily stored before commencing to transfer chunks thereof into the storage registers.

47. The method of claim 44, additionally comprising generating a redundancy code for each of the given number of sectors of data as the individual chunks of data are transferred from temporary storage into the storage registers, including using a common generating circuit for each of the given number of sectors of user data, separately storing intermediate results of the redundancy code generations in a separate code register for each of the given number of sectors of data and combining the stored intermediate results of one sector of data with a new quantity of data for the same sector.

48. The method of claim 47, additionally comprising including the redundancy code generated for each of the sectors of data in a final chunk of user data that is transferred to the storage registers.

49. The method of claim 44, additionally comprising maintaining a table of defective column addresses for each of the plurality of memory units, repetitively comparing destination addresses of the chunks of data with the column addresses in said table, and, in response to a positive comparison, inserting bits into the chunks prior to programming them into the memory blocks in a manner that the inserted bits are programmed into memory cells in defective columns.

50. The method of claim 44, wherein the memory cells within at least a plurality of said blocks are individually operated with more than two storage states in order to store more than one bit of data per memory cell.

51. The method of claim 44, wherein the memory cells within at least a plurality of said blocks are individually operated with exactly two storage states in order to store exactly one bit of data per memory cell.

52. A method of operating a re-programmable memory system having non-volatile memory cells organized into distinct blocks of a number of simultaneously erasable cells capable of storing a given quantity of data, the blocks of cells being further organized into a plurality of units, comprising: receiving and temporarily storing in a buffer memory at least a given number of sectors of user data to be programmed into the memory system, moving data in a stream from one of the given number of sectors of user data in the buffer at a time to a respective one of a given number of storage registers at a time, and thereafter moving the user data from the given number of storage registers in parallel to respective ones of a given number of memory cell blocks that are located within different ones of a given number of said units.

53. The method according to claim 52, wherein moving data from the storage registers to the memory cell blocks includes moving one chunk of a sector of user data from each of the given number of registers, wherein the amount of data in a chunk is equal to one half or less of the amount of data in a sector.

54. The method according to claim 53, wherein moving data from the buffer memory to the storage registers includes moving one chunk at a time

alternatively from the given number of sectors of user data stored in the buffer memory.

55. The method according to claim 52, wherein moving data from the storage registers to the memory cell blocks includes moving a full user data sector from each of the given number of registers.

56. The method according to claim 55, wherein moving data from the buffer memory to the storage registers includes moving data from one sector at a time in sequence from the given number of sectors of user data stored in the buffer memory.

57. The method according to claim 52, which additionally comprises, prior to commencing moving the stream of data, generating a data transformation bit field for each sector of user data and using that bit field to transform the user data that is moved in a stream, and further comprising inserting the generated transformation bit field into each of the given number of sectors of user data at its beginning.

58. The method according to claim 52, wherein moving data in a stream includes generating a redundancy code from the stream of user data of the individual sectors and appending the generated code to the ends of the user data from which they are generated.

59. The method according to claim 54, wherein moving data in a stream includes generating a redundancy code from the stream of user data of the individual sectors and appending the generated code to the ends of the user data from which they are generated, the redundancy code generation including separately storing intermediate results of the redundancy code generations in a separate code register for each of the given number of sectors of data and combining the stored intermediate results of one sector of data with a new quantity of data for the same sector.

60. The method according to claim 52, wherein moving data in a stream includes inserting bits into the data stream for storage within cells in any defective columns of the memory blocks, the inserted bits shifting the user data thereafter.

61. The method according to claim 54, wherein moving data in a stream includes inserting bits into the data stream for storage within cells in any defective columns of the memory blocks, the inserted bits shifting the user data thereafter, the bit insertion including referencing addresses of any defective columns within each of the given number of memory cell blocks into which user data is moved from the storage registers.

62. The method of any one of claims 52 - 57, wherein the memory cells within at least a plurality of said blocks are individually operated with more than two storage states in order to store more than one bit of data per memory cell.

63. The method of any one of claims 52 - 57, wherein the memory cells within at least a plurality of said blocks are individually operated with exactly two storage states in order to store exactly one bit of data per memory cell.

64. The method of any one of claims 52 - 57, wherein the memory cells within a first group of a plurality of said blocks are individually operated with more than two storage states in order to store more than one bit of data per memory cell, and wherein the memory cells within a second group of a plurality of said blocks different from said first group are individually operated with exactly two storage states in order to store exactly one bit of data per memory cell.

65. The method of any one of claims 52 - 57, wherein the sectors of user data stored in the memory cell blocks do not include characteristics of the memory cell blocks in which they are stored.

66. A memory system, comprising: an array of non-volatile memory cells formed into blocks of cells that are simultaneously erasable together, a plurality of data registers, a first data transfer circuit that moves data in parallel between the plurality of data registers and respective distinct blocks of the memory cell array, a buffer memory capable of storing a plurality of sectors of user data at the same time, a second data transfer circuit that moves user data in a stream between the buffer memory and one of the data registers at a time, a redundancy code circuit positioned in the path of the data stream to generate a redundancy code in real time from the data stream, and a defective column circuit positioned in the path of the data stream

to adjust the length of the stream to avoid defective columns within the memory cell array.

67. A method of operating a re-programmable non-volatile memory system having its memory cells organized into distinct blocks of simultaneously erasable cells, comprising: identifying a first group of said blocks for storing user data and a second group of said blocks for storing information of the characteristics of said first group of blocks, storing, in individual ones of the first group of said blocks, user data without data of the characteristics of said first group of blocks, and storing, in individual ones of the second group of said blocks, a plurality of records of characteristics of individual ones of the first group of blocks but without storing user data in the second group of blocks.

68. The method of claim 67, wherein storing the plurality of records in individual ones of the second group of said blocks includes storing a redundancy code generated from records written therein.

69. The method of claim 67, wherein storing, in the second group of blocks, the plurality of records of characteristics of the first group of blocks individually includes storing programming and reading characteristics of a corresponding one of the first group of blocks.

70. The method of claim 69, additionally comprising: reading the records from a plurality of said second blocks and storing the read records in a controller

memory, and when accessing one or more of the first group of blocks to program user data therein or to read user data therefrom, reading from the controller memory those of the records stored therein which contain the characteristics of said one or more of the first group of blocks being accessed.

71. The method of claim 70, wherein records of at least one of the plurality of second blocks stored in the controller memory which have the longest time since being read are removed therefrom when a limited capacity of the controller memory requires space to be made for records from another of the plurality of said second blocks to be stored therein in order to be read when one or more of corresponding ones of the first group of blocks is being accessed.

72. The method of claim 70, wherein, when a plurality of the first group of blocks with successive addresses are being accessed, an address of a record stored in the controller memory that corresponds to a first of the addressed block within the first group of blocks is calculated and remaining records within the controller memory that correspond to others of the plurality of the first group of blocks being accessed are addressed by incrementing from one record address to another.

73. The method of claim 67, wherein storing, in the second group of blocks, the plurality of records of characteristics of the first group of blocks individually includes storing an indication of whether a corresponding block within said first group is defective or not, and, if so, storing in the second group of blocks an

address of a substitute block, and, if not, storing in the second group of blocks operating characteristics of the corresponding block within said first group.

74. The method of claim 73, wherein storing operating characteristics includes storing any of programming, reading, erase or wear characteristics of the corresponding block within the first group.

75. The method of claim 67, wherein storing, in the second group of blocks, the plurality of records of characteristics of the first group of blocks individually includes storing indications of locations of any bad columns that extend through corresponding blocks within said first group.

76. The method of claim 75, wherein storing user data into individual ones of the first group of blocks and storing block characteristic records into individual ones of the second group of blocks includes skipping any bad column locations in the respective blocks.

77. The method of any one of claim 67, 69 or 73, wherein characteristics of the user data being stored in individual ones of the first group of said blocks are additionally stored therein along with the user data to which such characteristics relate.

78. The method of any one of claim 67, 69 or 73, wherein the method is practiced when the memory system is enclosed within a card that is removably connectable to a host system.

79. The method of any one of claim 67, 69 or 73, wherein the memory cells within at least a plurality of said blocks are individually operated with more than two storage states in order to store more than one bit of data per memory cell.

80. The method of any one of claim 67, 69 or 73, wherein the memory cells within at least a plurality of said blocks are individually operated with exactly two storage states in order to store exactly one bit of data per memory cell.

81. The method of claim 67, wherein storing the plurality of records in individual ones of the second group of said blocks includes storing the plurality of records to include individual values of a common set of operating characteristics of a plurality of the first group of blocks.

82. The method of claim 81, wherein the common set of operating characteristics of individual records includes any of programming, reading, erase or wear characteristics of a corresponding block within the first group of blocks.

83. The method of claim 81, wherein the method is practiced with the memory system enclosed within a card that is removably connectable to a host system.

84. A method of operating a re-programmable memory system having non- volatile memory cells organized into distinct blocks of simultaneously erasable cells, comprising: storing user data in a first group of a plurality of less than all of said blocks, and storing, in each of one or more others of said blocks in a second group, a plurality of records that each contains data of any one or more programming, reading, erasing or wear characteristics of a corresponding block within the first group of blocks.

85. The method of claim 84, wherein storing user data in the first group of blocks includes storing characteristics of the user data in the blocks containing the user data to which the user data characteristics pertain, and wherein characteristics of the user data are not stored in the second group of blocks.

86. The method of claim 84, wherein said data of programming, reading, erasing or wear characteristics are not stored in the first group of blocks, and further wherein said user data are not stored in the second group of blocks.

87. The method of claim 84, wherein the method is practiced with the memory system enclosed within a card that is removably connectable to a host system.

88. The method of claim 84, wherein the memory cells within at least a plurality of said blocks are individually operated with more than two storage states in order to store more than one bit of data per memory cell.

89. A method of operating a re-programmable non-volatile memory system having its memory cells organized into distinct blocks of simultaneously erasable cells, comprising: designating a first group of said blocks for storing user data and a second group of said blocks for storing information of the characteristics of said first group of blocks, storing, in individual ones of the first group of said blocks, user data plus characteristics of the user data being written therein, and storing, in individual ones of the second group of said blocks, a plurality of records of characteristics of individual ones of the first group of blocks.